LB

3

Unit

Cascading Style Sheets(CSS)

Compiled by:
LB Gurung
lbgurung00@gmail.com

CSS: Introduction

- ❑ CSS is a W3C(World Wide Web Consortium) standard for describing the presentation or appearance of HTML elements. **Or**
- ❑ CSS, or Cascading Style Sheets, is a styling language used to enhance the visual presentation of HTML documents. It allows user to control the layout, colors, and layout of their website.
- ❑ CSS consists of various styles which defines how to display HTML elements so as to make the design of the website **dynamic** and attractive .
- ❑ Styles are normally stored in Style Sheets.

CSS: Introduction

- ❑ Styles are normally stored in Style Sheets.
- ❑ With CSS, We can design:

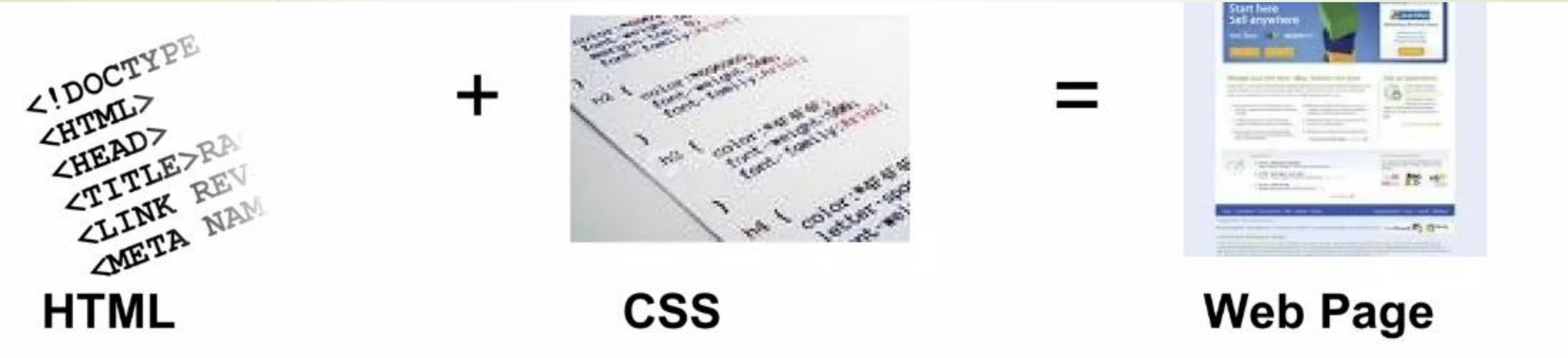
  - font properties,

  - colors,

  - sizes

  - borders, backgrounds images and even the position of elements and many more..

CSS: Introduction



Summary:

Cascading: **Falling of Styles**

Style: Adding designs/Styling our HTML tags

Sheets: Writing our style in different documents

Why to use Cascading Style Sheets(CSS)?

By using CSS, we can make more affecting, attractive and customized webpages in less time and code

- CSS Saves times
- Offline Browsing
- Easy maintenance
- Responsive Design
- Animation and Effects
- Ease of Updates

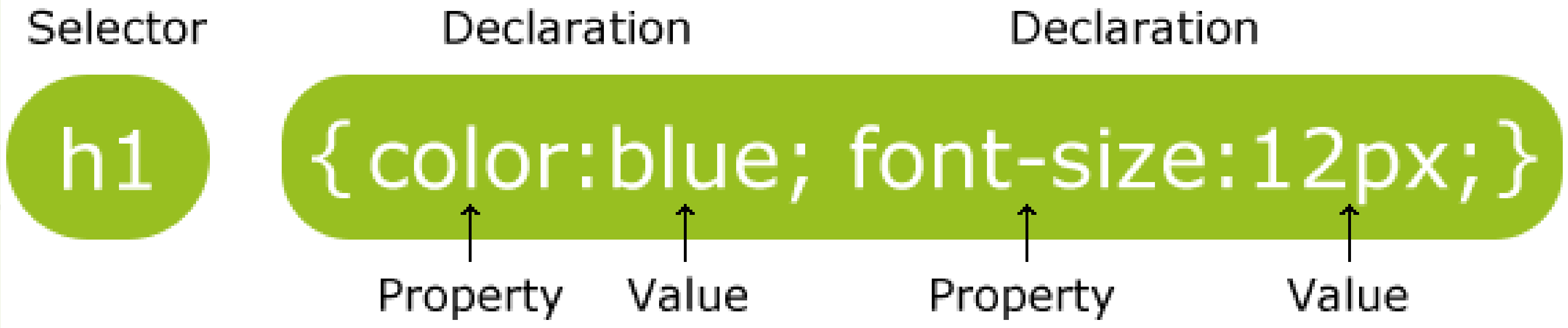
LB

❑ Cascading Style Sheets(CSS) editors:

- ✓ Atom
- ✓ Visual Studio Code
- ✓ Brackets
- ✓ Espresso(For Mac OS User)
- ✓ Notepad++(Great for HTML & CSS)
- ✓ Komodo Edit (Simple)
- ✓ Sublime Text (Best Editor)

❑ CSS Syntax:

A CSS rule-set consists of a selector and a declaration block:



- ✓ The selector points to the HTML element you want to style.
- ✓ The declaration block contains one or more declarations separated by semicolons.
- ✓ Each declaration includes a CSS property name and a value, separated by a colon.
- ✓ A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
   p {
    color: red;
   text-align: center;
}
</style>
</head>
<body>
<p>BIT 4rth Semester </p>
<p>This paragraph is styled with CSS.</p>
</body>
</html>
```

Advantages of using CSS

- ✓ Web pages will load faster
- ✓ It makes webpages browser compatible with almost all the browsers
- ✓ Future redesigns will be more efficient
- ✓ It allows more interactive style elements, including font, font size, font color, which can make pages more usable for people with disabilities
- ✓ It can help to make webpages available for different media like desktop PC , Cell phone etc.

- ❑ How many ways we can do declare CSS in HTML?

  (types of CSS)
- ❑ What are the different types of CSS used in webpage designing? Explain
- ❑ Describe the CSS layout techniques.

- CSS is added to HTML pages to format the document according to information in the style sheet.
- There are three ways to insert CSS in HTML document:

a. Inline CSS

b. Internal CSS

c. External CSS

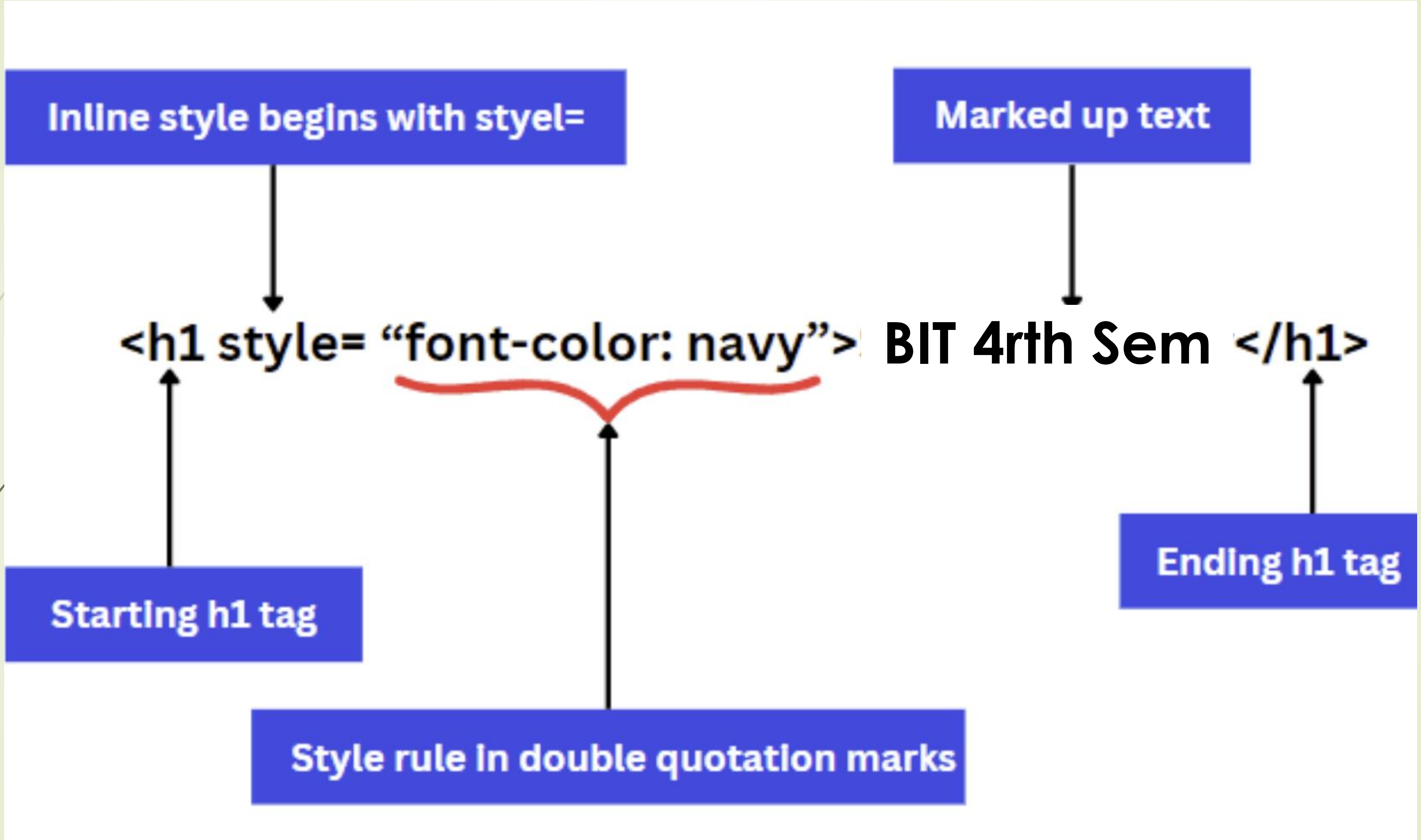
a. Inline CSS

- ✓ It is used to apply CSS on a single line of element
- ✓ For Inline CSS every style content is in HTML elements.
- ✓ It is used for a limited section. Whenever our requirements are very small we can use inline CSS.

```
<P style="color:red;">Welcome</P>
```

a. Inline CSS

✓ For example:


Inline style begins with styel=
Marked up text
Starting h1 tag
Ending h1 tag
Style rule in double quotation marks

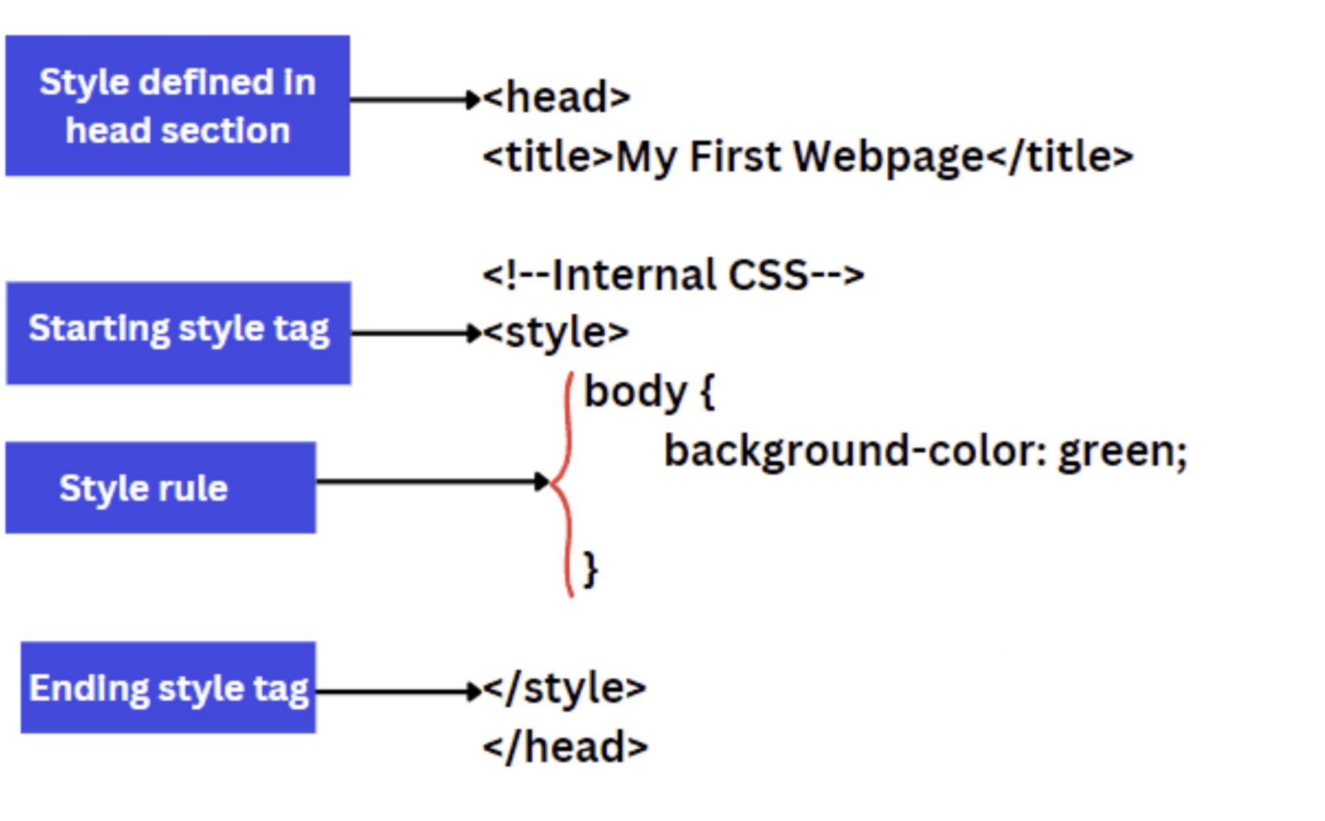
```
<h1 style= “font-color: navy”> BIT 4rth Sem </h1>
```

Fig: An example of Inline CSS

b. Internal CSS

- ✓ It is used to apply CSS on a single document of page. It can affect all the element of the page.
- ✓ It is written inside the style tag within head section of html
- ✓ For example:

```
<!doctype html>
<html>
<head>
<style>
Body{background-color:blue;}
h1{color:red;}
</style>
</head>
<Body>
<H1>Welcome 24 Months IT</H1>
</body>
</html>
```



```
<head>
<title>My First Webpage</title>

<!--Internal CSS-->
<style>
   body {
        background-color: green;

   }

</style>
</head>
```

Fig: An example of Internal CSS style sheet

c. External CSS

- ✓ It is used to apply CSS on multiple pages or all pages.
- ✓ It can be written in any text editor
- ✓ The file should not contain any html tags.
- ✓ The file must be saved with .css extension
- ✓ For example:

```
Body {
Background-color:blue;
 }
H1{
Color:pink;
}
```

Note: Save the code to any file name having an extension .css

c. External CSS

```
<!doctype html>
<html>
<head>
<link rel="stylesheet"
      type="text/css"
      href="style.css">
</head>
<body>
<h1>This is heading</h1>
</body>
</html>
```

- ❑ Which is Commonly used in CSS?

  - ✓ All three methods of adding CSS styles inline, external, and internal are commonly used in CSS, depending on the specific needs of the website or web application. However, **external CSS** is the most commonly used method because it allows for the separation of content and presentation, making it easier to maintain and update the styles across multiple pages.

- ❑ What is CSS selector? Explain the different types of selector in css with examples.

- ✓ A CSS selector is the part of a CSS rule set that actually selects the content you want to style

Types of CSS selector

The different types of CSS selectors are:

a. Tag selector/Element Selector

b. Class selector

c. ID selector

d. Group selector

e. Contextual selector

   i. Descendent selector

   ii. Child Selector

   iii. Adjacent sibling selector

   iv. General sibling selector

f. Attribute selector

g. Pseudo classes

h. Pseudo elements

i. Universal Selector

a. Tag selector:

- ✓ To select tags by their name and apply styles on them we use tag selector
- ✓ It selects every html tag with the specified tag name and applies styles on them.
- ✓ Syntax:

```
selector
 {
   declaration list;
}
```

Note: to implement tag selector, in place of selector , we write tag name

a. Tag selector:

✓ Example:

```
P
{
  color:green;
  font-size:16pt;
  border:2px solid red;
}
```

Declaration list

Note: It selects every P tag available on the page and applies specified on them

```
<!doctype html>
<html>
<head>
<title> tag selector demo</title>
<style>
p
{
color:red;
Font-size:14pt;
}
</style>
</head>
<body><h4>HTML</h4>
<p>
HTML is a simple, small and powerful formatting language for webpages.
HTML is a markup language containing set of tags understood by browsers.
HTML stands for Hyper text markup language
</p>
<p>
CSS is a simple, small and powerful formatting language for webpages.
CSS is a presentational  language containing set of styles understood by browsers.
CSS stands for Cascading Style Sheet
</p>
</body>
</html>
```

Class selector:

- ✓ To select tags by their class attribute value and apply styles on them we use class selector
- ✓ It selects every html tags available on the page with a specified class attribute value and applies specified styles on them
- ✓ You can specify same class attribute value to more than one tag. If required

Class selector:

- ✓ Syntax:

```
 selector
  {
     declaration list;
}
```

Note: To implement class selector, in place of selector , we write class attribute value preceded by period (.) or dot.

- ✓ Syntax of class selector:

```
        <tagname class="classattributevalue">
         .classattributevalue
          {
                declaration list;
}
```

Class selector:

- ✓ Example:

```
<p class=“solidborder”>
 .solidborder
{
   border=2px solid red;
}
```

```
<!doctype html>
<html>
<head>
<title> tag selector demo</title>
<style>
p
{
Border:2px solid red;
Font-size:14pt;
}
.redborder
{
 border:2px solid red;
}
</style>
</head>
<body><h4>HTML</h4>
<p>
Some text
</p>
<p class=“redborder”>
Some text
</p>
<p class=“redborder”>Some text
</p>
</body>
</html>
```

ID Selector:

- ✓ To select tags by their “id attribute value” and apply styles on them we use id selector.
- ✓ In CSS # Symbol indicates ID selector
- ✓ Syntax:

```
<tagname id=“idattributevalue”>
#idattributevalue
{
Declaration list;
}
```

- ✓ It selects every html tag which has specified id attribute value and applies styles on them

ID Selector:

- ✓ Example:

```
<h1 id="solidborder">
#solidborder
{
        border:2px solid red;
}
```

ID Selector:

```
<!doctype html>
<html>
<head>
<title>demo of id selector</title>
<style type="text/css">
#heading1
{
          border:2px solid red;
}
</style>
</head>
<body>
          <h1 id="heading1"> Heading 1 text</h1>
          <p>Some text</p>
          <p>Some text</p>


</body>

</html>
```

Group Selector:

- ✓ There are situations where different tags have similar styles , to reduce code redundancy we create group selector
- ✓ To implement group selector , in place of selector we write comma separated list of selectors
- ✓ Syntax:

```
selector1 , selector2, selector3
{
declaration list;
}
```

Group Selector:

- ✓ Example:

```
h3
{
background-color:gray;
color:white;
border:2px solid black;
}
 p
 {
   background-color:gray;
   color:white;
   border:2px solid black;
}
 h3,p
{
background-color:gray;
color:white;
 border:2px solid black;
}
```

Note: Comma list of selectors is known as Group selector

Group Selector:

```
<!doctype html>
<html>
<head>
<title>demo of group selector</title>
<style type="text/css">
h3,p
{
 background-color:gray;
 color:white;
  font-size:18pt;
}
</style>
</head>
<body>
            <h3> heading text</h3>
            <p> Paragraph text</p>


</body>

</html>
```

Universal Selector:

- ✓ It is used to target any html element on page and apply styles on it
- ✓ Syntax:

```
 *
{
 declaration list;
}
```

Example:

```
 *
{
 border:2px solid red;
}
```

Note: It selects every html elements on the page and applies 2px solid red border.

Universal Selector(*):

- ✓ It is used to target any html element on page and apply styles on it
- ✓ Syntax:

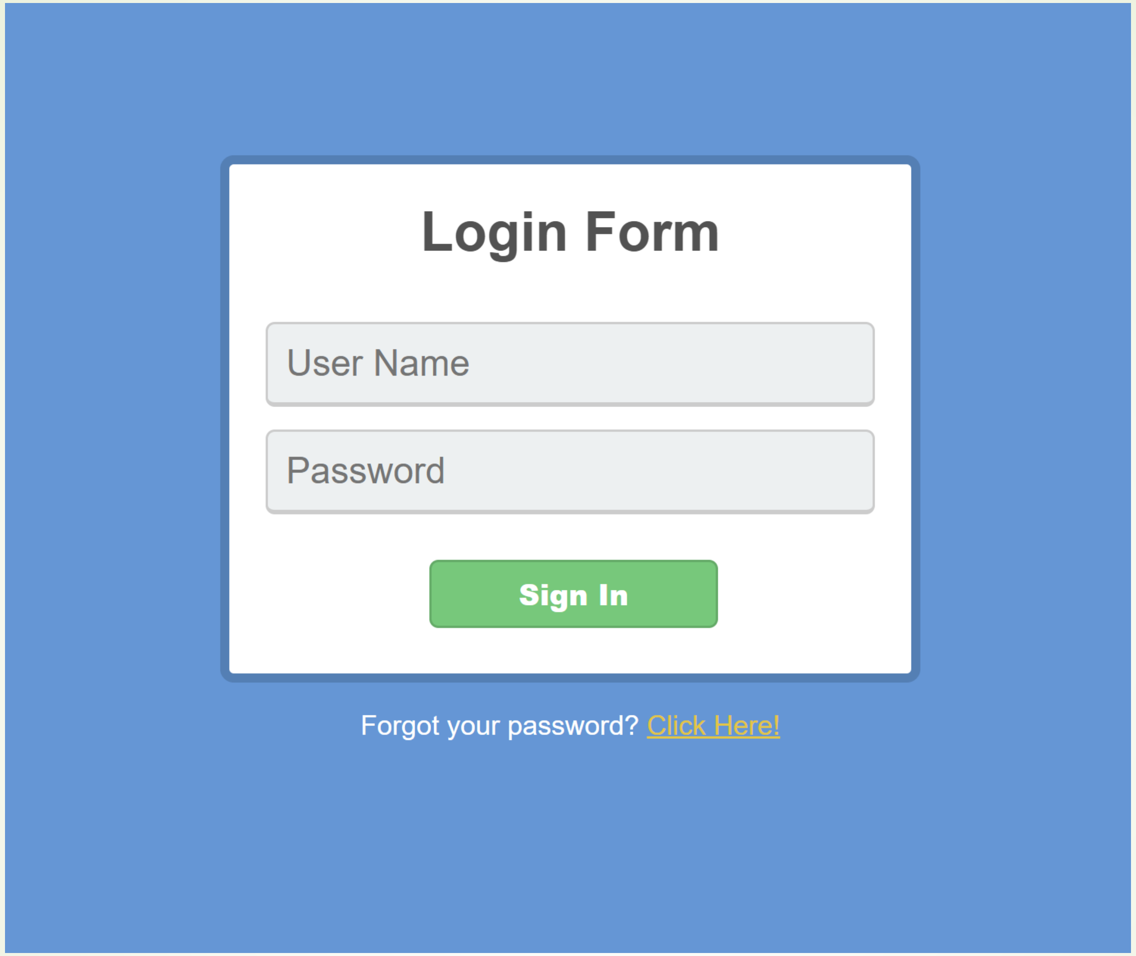
```
 *
{
 declaration list;
}
```

Example:

```
 *
{
 border:2px solid red;
}
```

Note: It selects every html elements on the page and applies 2px solid red border.

LAB Work:



Login Form

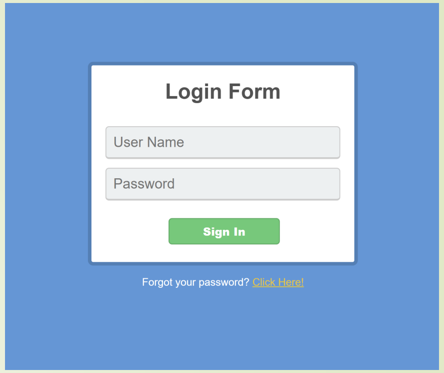
User Name

Password

Sign In

Forgot your password? Click Here!

LAB Work:



Forgot your password? Click Here!

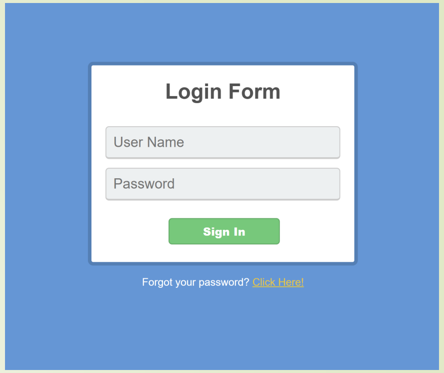
```
<!DOCTYPE html>
<html>
<head>
    <title>Simple Login Form</title>
    <link rel="stylesheet" href="form1.css">

</head>
<body>
    <div id="absoluteCenteredDiv">
        <form action="index.html" method="post">
            <div class="box">
                <h1>Login Form</h1>
                <input class="username" name="username" type="text"
placeholder="User Name">
                <input class="username" name="username" type="password"
placeholder="Password">
                <a href="#"><div class="button">Sign In</div></a>
            </div>
        </form>
        <p>Forgot your password? <a class="fpwd" href="#">Click Here!</a></p>
    </div>
</body>
</html>
```

Note: Save this code with extension .htm

```
body{
  font-family: 'Open Sans', sans-serif;
  background:#3498db;
  margin: 0 auto 0 auto;
  width:100%;
  text-align:center;
  margin: 20px 0px 20px 0px;
}
p{
  font-size:12px;
  text-decoration: none;
  color:#ffffff;
}
h1{
  font-size:1.5em;
  color:#525252;
}
.box{
  background:white;
  width:300px;
  border-radius:6px;
  margin: 0 auto 0 auto;
  padding:0px 0px 70px 0px;
  border: #2980b9 4px solid;
}
.username{
  background:#ecf0f1;
  border: #ccc 1px solid;
  border-bottom: #ccc 2px solid;
  padding: 8px;
  width:250px;
  color:#AAAAAA;
  margin-top:10px;
  font-size:1em;
  border-radius:4px;
}
```

```
.button{
    background:#2ecc71;
    width:125px;
    padding-top:5px;
    padding-bottom:5px;
    color:white;
    border-radius:4px;
    border: #27ae60 1px solid;
    margin-top:20px;
    margin-bottom:20px;
    float:left;
    margin-left:88px;
    font-weight:800;
    font-size:0.8em;
  }
  .button:hover{
    background:#2CC06B;
  }
  .fpwd{
     color:#f1c40f;
     text-decoration: underline;
  }
  #absoluteCenteredDiv{
     position: absolute;
     top:0;
     bottom: 0;
     left: 0;
     right: 0;
     margin: auto;
     width:400px;
     height: 300px;
     text-align: center;
  }
```

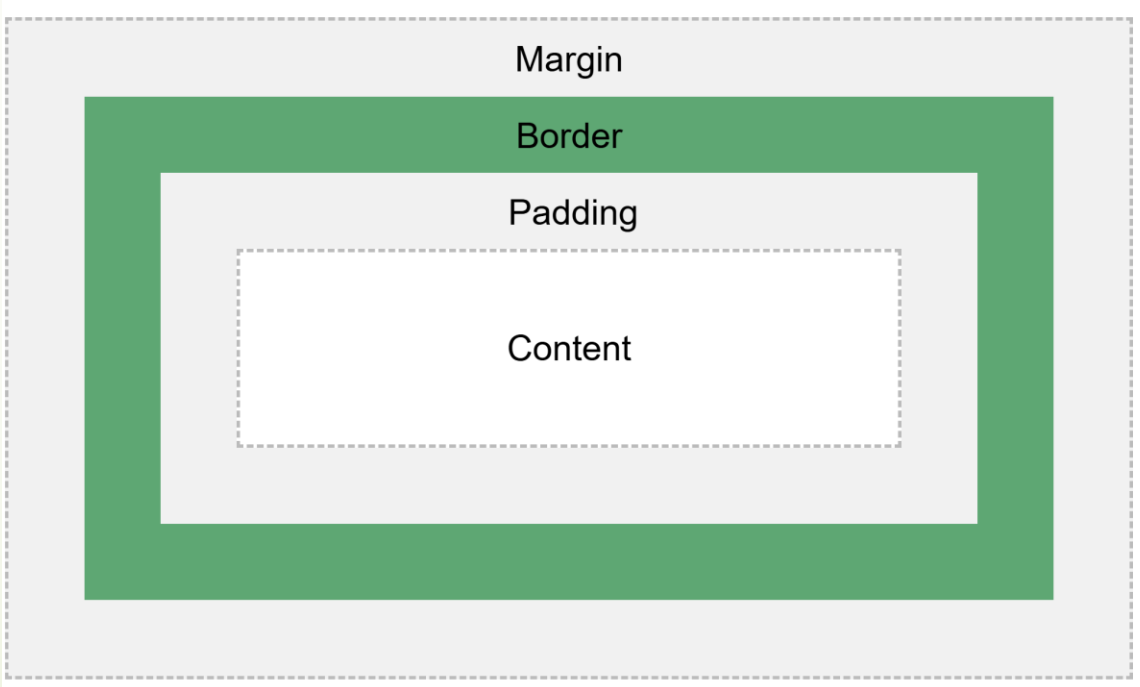


Forgot your password? Click Here!

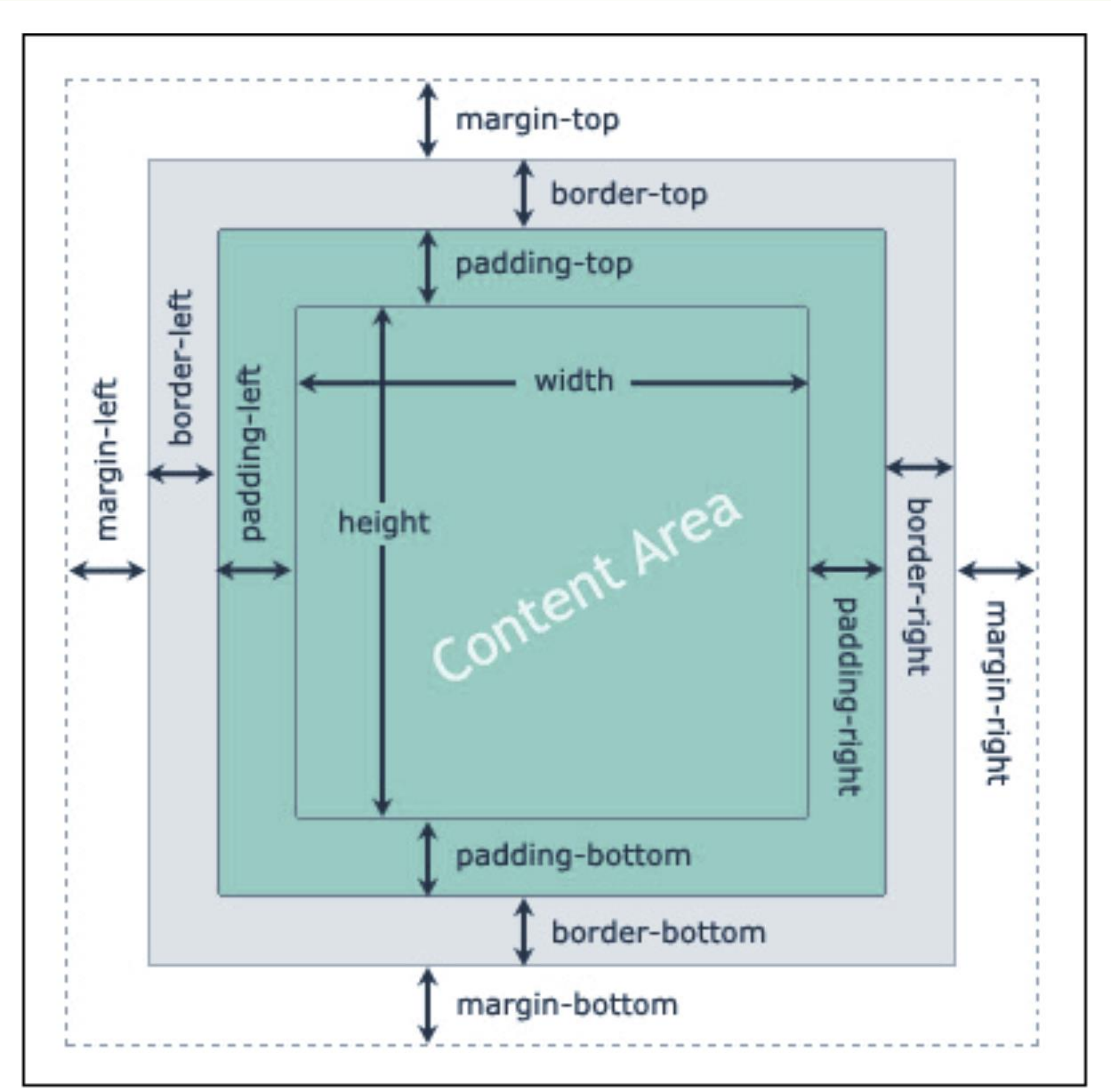
Note: Save this code as **form1.css**

The CSS Box Model

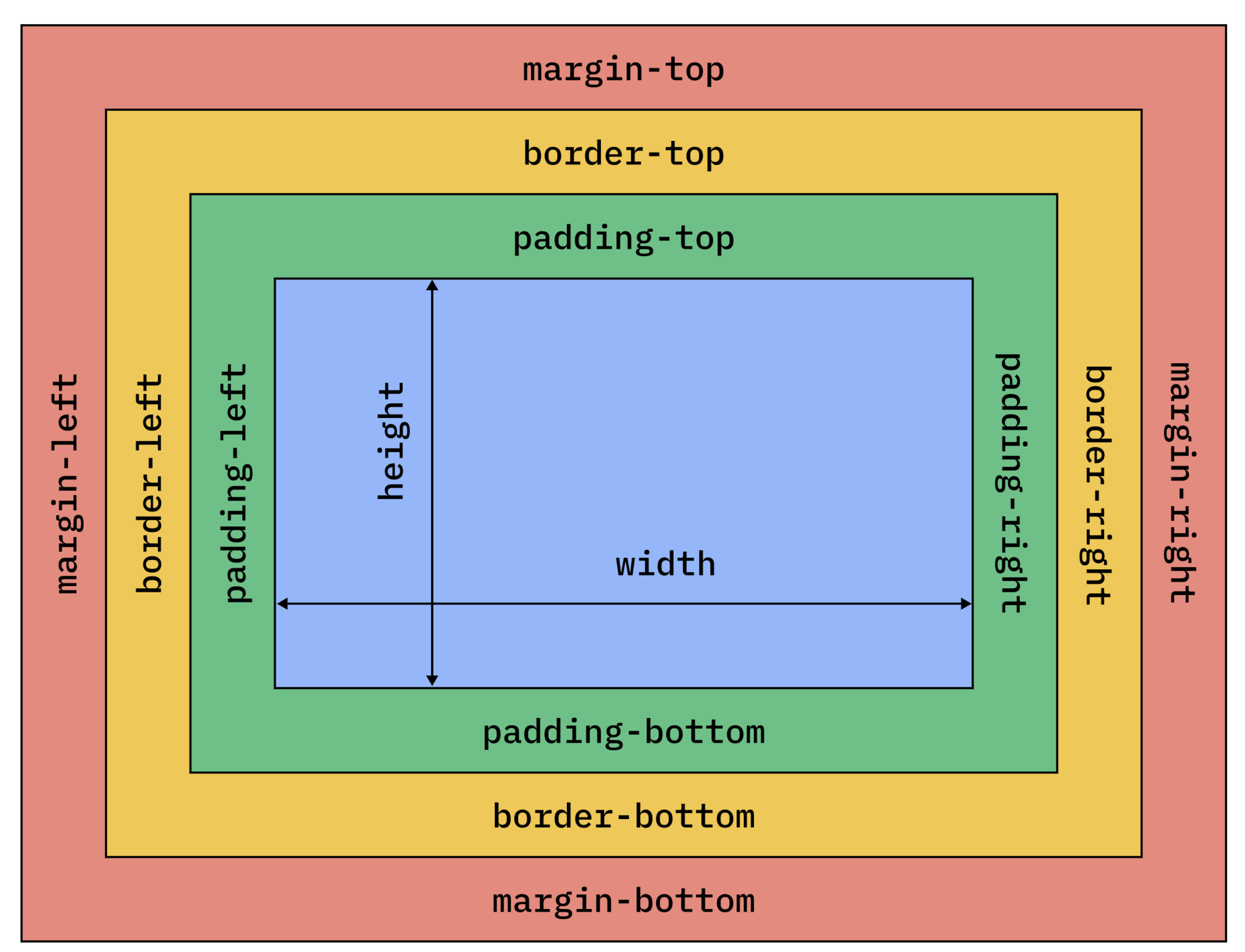
- ✓ **CSS is used for the layout and presentation of the HTML document.** The CSS box model is a fundamental concept that defines how the element's dimensions and spacing are calculated. The box model treats every HTML element as a rectangular box consisting of **content, padding, border, and margin.**



The CSS Box Model

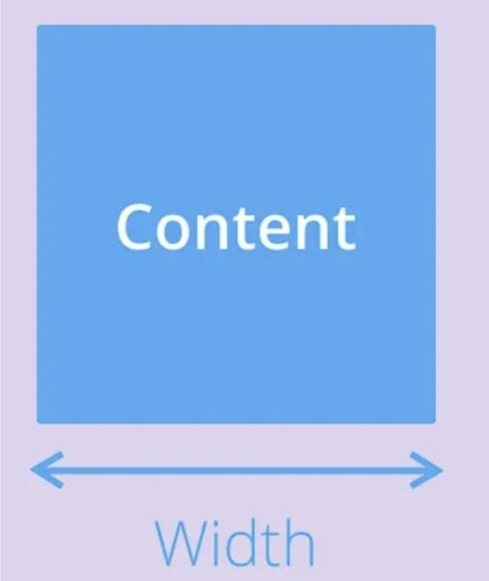


The CSS Box Model



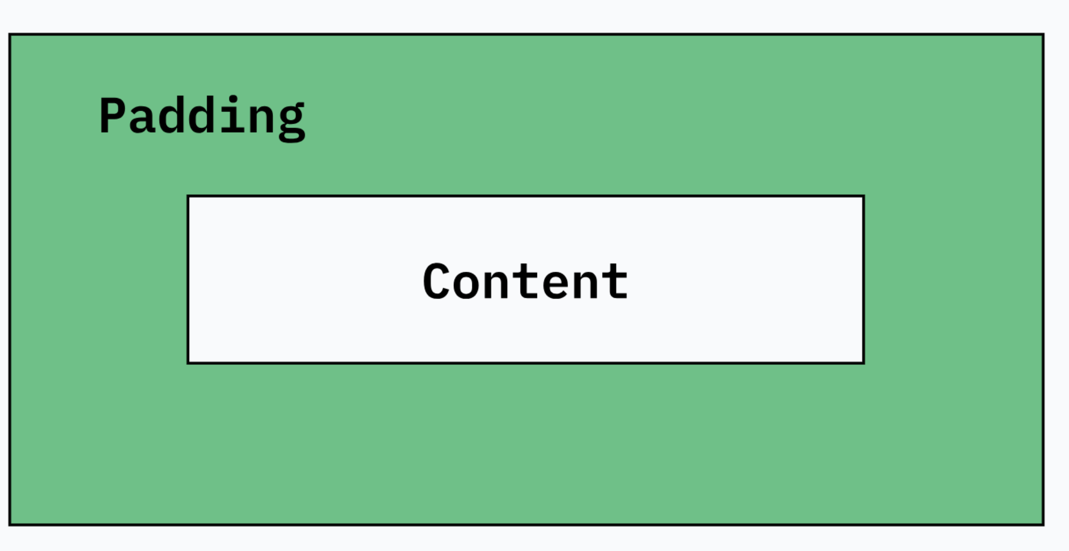
Content:

The element's content forms the innermost box. The width and height properties set the dimensions of the content box, like this, width: 100px; height: 200px;, which sets the content box to 100px by 200px. This area contains the text or image for the element.



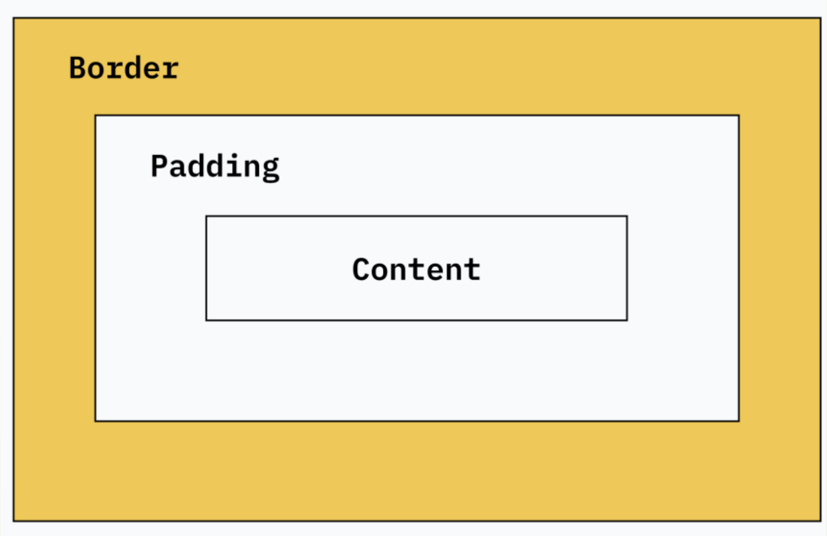
Padding :

- ✓ Transparent space between the content and the border of an element. Padding adds extra space between the content area and its border. It has the same background as the content itself.
- ✓ Example CSS property: padding: 20px; (this will add 20px of padding on all sides).



Border :

- ✓ The border wraps the padding and the content. It is the edge of the element.
- ✓ Borders can have different styles (solid, dotted, dashed), thicknesses, and colors.
- ✓ Example CSS property: border: 2px solid black;.



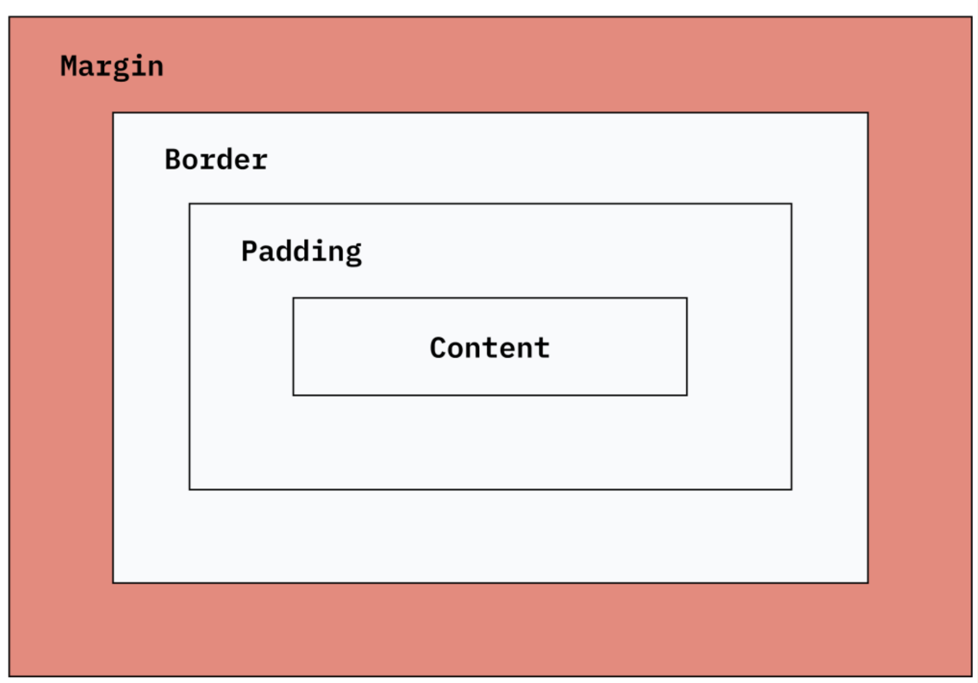
Border

Padding

Content

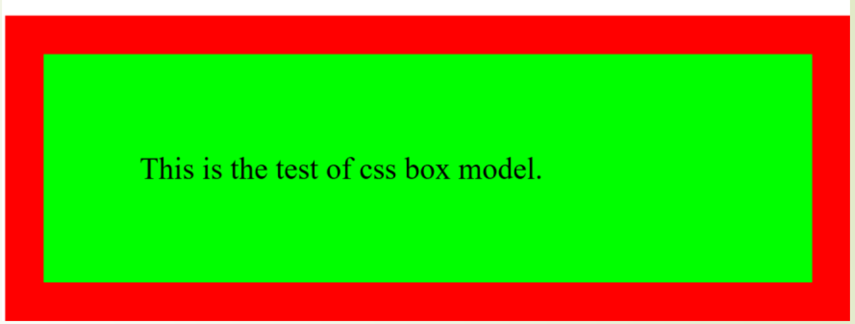
Margin :

- ✓ Margin is the outermost layer, creating space between the element and neighboring elements.
- ✓ Like padding, you can control the margin on each side separately or use a single value for all sides. Example CSS property: margin: 15px;



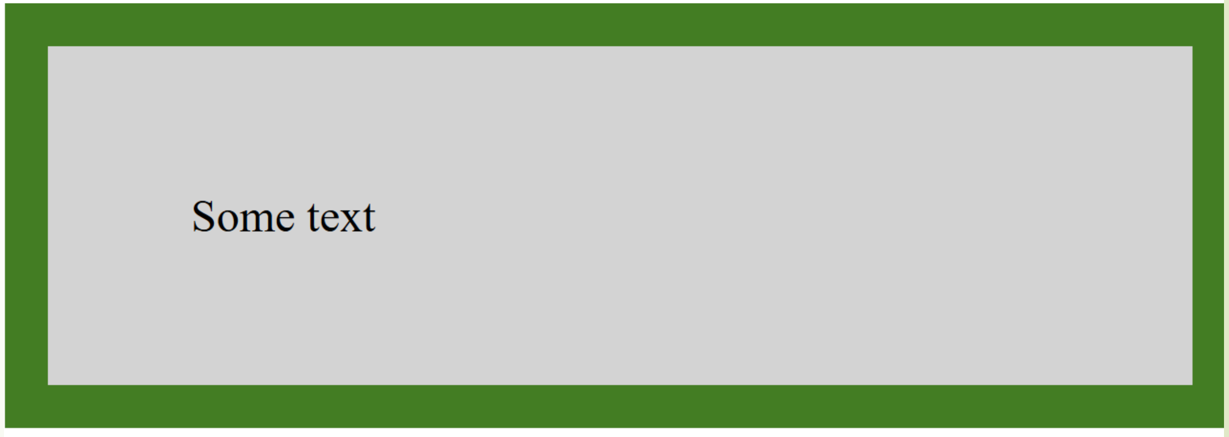
Lab Work:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>
<div>
Some text
</div>
</body>
</html>
```



This is the test of css box model.

Output:



Some text

HTML div tag:

- ✓ The <div> tag in HTML is a block-level element used as a container to group other HTML elements. It is widely used with CSS and JavaScript to structure a webpage and apply design and interactivity.
- ✓ Mostly <div> are used for divide complete page into section like footer, header, left, right section.

HTML div tag:

Example:

```
<!doctype html>
<html>
<head>
<style>
  .header {
    width: 200px;
    height: 100px;
    background-color: lightblue;
    border: 2px solid blue;
    padding: 10px;
  }
</style>
</head>
<body>
<div class="header">Header Content</div>
<div class="content">Main Content</div>
<div class="footer">Footer Content</div>

</body>
</html>
```

Note: Here, each <div> creates a separate block of content that can be styled differently.

CSS position(Position property)

- ✓ Positioning refers to how elements are placed within the layout.

The three main types of positioning

- **relative**,
- **float**, and
- **Absolute**

Other are static , sticky

are different techniques for controlling the layout of elements on a webpage.

CSS position(Position property)

- **Relative Position:** This position places the element relative to its normal position. This position is relative to normal document flow. This property works the same as a static
- Here top/right/bottom/left properties can be applied to elements.
- Example:

CSS position(Position property)

```
<!DOCTYPE html>
<html>
<head>
<style>
  .box {
    width: 100px;
    height: 100px;
    background-color:Lime;
    position: relative;
    top: 30px;
    left: 30px;
  }
</style>
<div class="box">I'm a relatively positioned box</div>
</body>
</html>
```

CSS position(Position property)

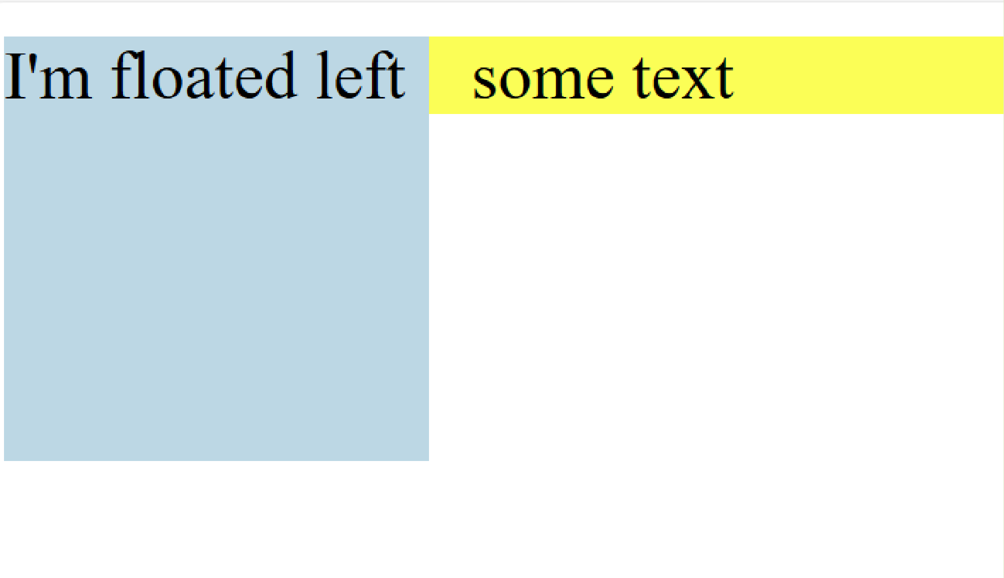
- **Float Position:** The float property is used to make elements float to the left or right within their containing block, allowing text and inline elements to wrap around them. This property is often used to create multi-column layouts.
- Example:

CSS position(Position property)

```
<style>
 .left-box {
   width: 100px;
   height: 100px;
   background-color: lightblue;
   float: left;
   margin-right: 10px;
 }

 .content {
   background-color: yellow;
 }
</style>

<div class="left-box">I'm floated left</div>
<div class="content">
 some text
</div>
```



I'm floated left some text

CSS position(Position property)

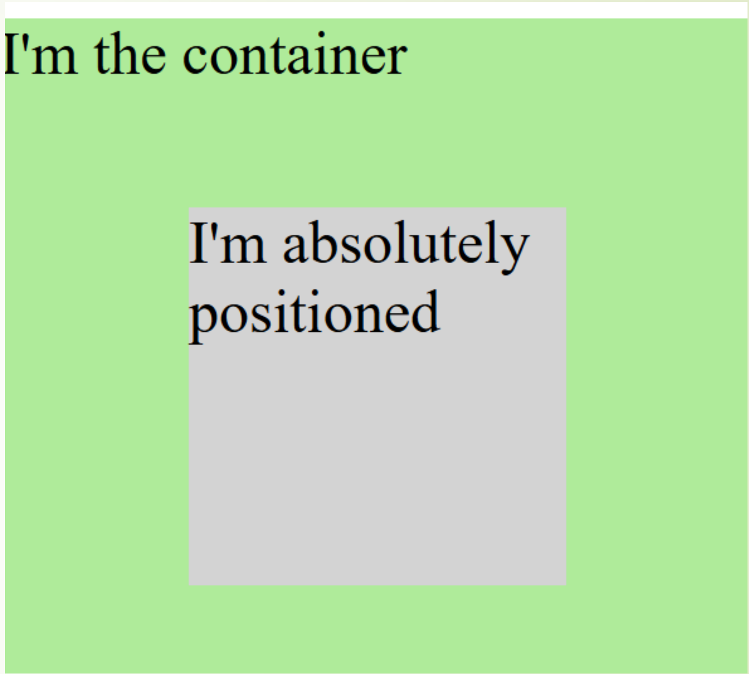
- **Absolute Position:** In this position, an element places itself with respect to itself and excludes itself from the document’s normal flow. Generally, they position themselves in the top-left of their ancestor or parent element.

CSS position(Position property)

- **Example:**

```
<style>
  .container {
    position: relative;
    width: 200px;
    height: 200px;
    background-color: lightgreen;
  }

  .absolute-box {
    position: absolute;
    top: 50px;
    left: 50px;
    width: 100px;
    height: 100px;
    background-color: lightgray;
  }
</style>
<div class="container">
  I'm the container
  <div class="absolute-box">I'm absolutely positioned</div>
</div>
```



I'm the container

I'm absolutely positioned